BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 4, June 2026 | Hal 1329-1342
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i4.976

Analisis Komparasi Convolutional Neural Network dan Learning Vector Quantization dalam Klasifikasi Khat Arab Digital

Sabri T Rahman[*], Yuhandri, Sumijan

Fakultas Ilmu Komputer, Program Studi Magister Teknik Informatika, Universitas Putra Indonesia YPTK Padang, Padang, Indonesia
Email: [1,*]sabriupiyptk@gmail.com, [2]yuhandri.yunus@gmail.com, [3]sumijan@upiyptk.ac.id
Email Penulis Korespondensi: sabriupiyptk@email.com

Abstrak−Khat Arab merupakan tulisan yang memiliki karakteristik visual kompleks, seperti variasi bentuk huruf, ketebalan goresan, tekstur, serta perbedaan gaya penulisan. Kompleksitas tersebut menimbulkan kesulitan saat pengenalan jenis khat secara manual. Penelitian ini bertujuan untuk menganalisis dan membandingkan kinerja metode *Convolutional Neural Network* (CNN) dan *Learning Vector Quantization* (LVQ) dalam mengklasifikasikan lima jenis citra digital khat Arab, yaitu Diwani, Farsi, Naskh, Ruqaa, dan Tuluth. Dataset diperoleh dari platform *Kaggle.com.* Arsitektur CNN terdiri dari *input layer* 100×100×1, dua *convolutional layer* masing-masing 32 dan 64 *filter* berukuran 3×3, diikuti aktivasi *ReLU* dan *max pooling* dengan *stride* 2, *fully connected layer* 64 *neuron*, bagian akhir *fully connected layer* sesuai jumlah kelas, *softmax layer*, dan *classification layer*. *Training* CNN menggunakan *K-fold Cross Validation* nilai K-5, menerapkan *data augmentation* setiap *fold*. LVQ menggunakan *Local Binary Pattern* (LBP) untuk ekstraksi fitur citra 100×100, parameter radius 1, *neighbors* 8, *cell size* [48 48], dan normalisasi L2. Hasil ekstraksi untuk *training* dengan inisialisasi *prototype* total 25 dari 5 kelas. Proses menggunakan *K-fold Cross Validation* nilai K=5. Dari 40 data uji, model CNN mendapatkan akurasi 87,5%, sedangkan akurasi model LVQ 85%. Algoritma CNN menunjukkan performa yang lebih baik menangani kompleksitas pola visual khat Arab. Sementara itu, LVQ memiliki keunggulan dalam kesederhanaan arsitektur dan efisiensi komputasi. Penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan sistem klasifikasi citra khat Arab serta menjadi referensi dalam pemilihan metode yang optimal untuk pengenalan khat Arab.

Kata Kunci: Khat Arab; Klasifikasi Citra; CNN; LVQ; LBP

Abstract−Arabic khat is a form of writing that possesses complex visual characteristics, such as variations in letter shapes, stroke thickness, texture, and stylistic differences. This complexity creates challenges in manually recognizing different types of khat. This study aims to analyze and compare the performance of Convolutional Neural Network (CNN) and Learning Vector Quantization (LVQ) methods in classifying five types of Arabic khat digital images, namely Diwani, Farsi, Naskh, Ruqaa, and Tuluth. The dataset was obtained from the Kaggle.com platform. CNN architecture consists of an input layer of 100×100×1, followed by two convolutional layers with 32 and 64 filters of size 3×3, each followed by ReLU activation and max pooling with stride 2. The network then includes a fully connected layer with 64 neurons, a final fully connected layer corresponding to the number of classes, a softmax layer, and a classification layer. CNN training was conducted using 5-fold cross-validation, applying data augmentation in each fold. For the LVQ method, Local Binary Pattern (LBP) was used for feature extraction from 100×100 images with parameters: radius 1, 8 neighbors, cell size [48 48], and L2 normalization. The extracted features were used for training with an initialization of 25 prototypes from 5 classes. The process also employed 5-fold cross-validation. From 40 testing samples, the CNN model achieved an accuracy of 87.5%, while the LVQ model achieved an accuracy of 85%. The CNN algorithm demonstrated better performance in handling the complex visual patterns of Arabic khat. Meanwhile, LVQ showed advantages in architectural simplicity and computational efficiency. This research is expected to contribute to the development of Arabic khat image classification systems and serve as a reference in selecting optimal methods for Arabic khat recognition.

Keywords: Arabic Khat; Image Classification; CNN; LVQ; LBP

1. PENDAHULUAN

Khat merupakan seni kaligrafi yang menggunakan huruf Arab sebagai unsur utama dalam pembuatannya, juga bentuk seni rupa tulis yang menekankan pada keindahan, keharmonisan, dan proporsi huruf sehingga menjadi ekspresi estetika dan spiritual dalam budaya Islam [1]. Masih banyak yang kurang memahami kaidah penulisan kaligrafi Arab dan beberapa diantaranya belum dapat membedakan jenis-jenis khat yang ada [2]. Namun karakteristik citra khat berupa garis kurva kompleks, variasi ketebalan goresan, tekstur kertas, noise manuskrip, serta variasi skala dan orientasi menimbulkan tantangan tersendiri dibandingkan pengenalan font cetak biasa.

Perkembangan teknologi *Artificial Intelligent* (AI) yang menjembatani kesenjangan antara kemampuan manusia dengan mesin telah berkembang pesat pada era modern ini. Salah satunya bidang *computer vision*, yang membuat mesin mampu melihat dunia dengan cara yang sama seperti manusia [3]. Teknologi ini membantu manusia dalam mengolah dan memperoleh informasi dari data visual. Salah satu implementasi pemrosesan citra digital pada teknik *computer vision* adalah klasifikasi citra (*image classification*) yang bertujuan untuk memberikan label atau kategori pada sebuah gambar berdasarkan isi visualnya. Kunci utama dalam klasifikasi citra adalah ekstraksi fitur, yaitu proses mengidentifikasi dan merepresentasikan ciri visual penting dari data gambar mentah sehingga bisa dianalisis dan diklasifikasikan dengan lebih efektif [4]. Dalam pemanfaatan teknologi *computer vision*, proses pengenalan dan pengelompokan jenis khat Arab dapat dilakukan dengan lebih mudah. Agar komputer mampu menjalankan tugas secara efektif, diperlukan pelatihan model menggunakan algoritma klasifikasi citra untuk mempelajari ciri khas dari setiap jenis khat. Dengan cara ini, sistem dapat mengidentifikasi dan membedakan jenis-jenis kaligrafi Arab secara akurat, sehingga memudahkan mereka yang tidak memiliki pengetahuan mendalam tentang kaligrafi, untuk mengenali setiap jenis khat Arab dengan cepat dan efisien [5].

Beberapa algoritma yang digunakan untuk klasifikasi citra diantaranya adalah *Convolutional Neural Network* (CNN) dan *Learning Vector Quantization* (LVQ). CNN telah menjadi jaringan saraf tiruan yang paling representatif

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 4, June 2026 | Hal 1329-1342
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i4.976

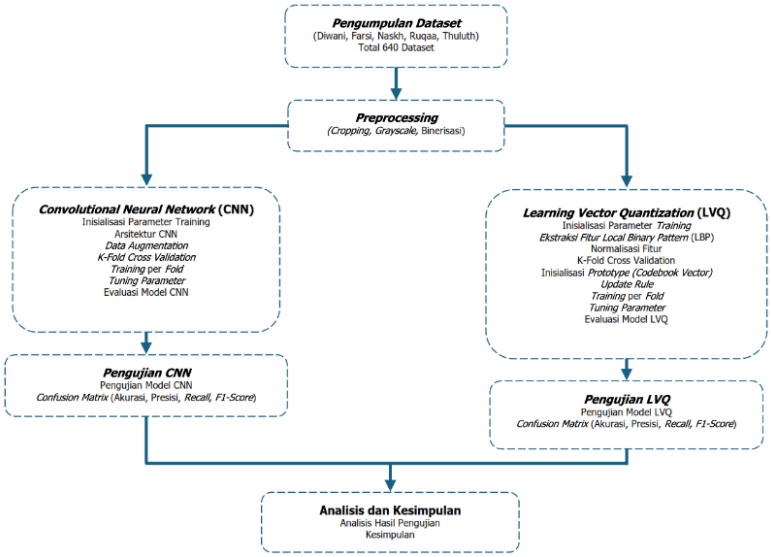
dalam *deep learning*. CNN telah dimanfaatkan untuk menyelesaikan tugas-tugas visual yang rumit dengan kebutuhan komputasi tinggi dan terutama digunakan dalam klasifikasi citra, segmentasi, deteksi objek, pemrosesan video, pemrosesan bahasa alami, serta pengenalan suara [6]. *Learning Vector Quantization* (LVQ), sejak diperkenalkan oleh Kohonen pada Tahun 1990, telah menjadi salah satu keluarga algoritma *supervised learning* yang penting. Pada fase pelatihan, algoritma ini menentukan prototipe yang mewakili kelas-kelas dalam data yang diberikan. Prediksi terhadap sampel baru dilakukan berdasarkan bidang reseptif (*receptive fields*) dari prototipe. Dengan kata lain, sebuah sampel baru diklasifikasikan dengan menghitung jarak dari sampel tersebut ke semua prototipe, kemudian memberikannya label dari prototipe yang jaraknya paling dekat. Perhitungan prototipe dan penentuan bidang reseptifnya dapat dilakukan dengan berbagai cara [7].

Beberapa penelitian yang telah dilakukan diantaranya menggunakan CNN untuk pengenalan tulis tangan aksara arab, perbandingan data uji dan data latih 70:30, serta akurasi yang diperoleh adalah 78,10% [8]. Penelitian menggunakan LVQ untuk pengenalan jenis teks kaligrafi. Ada enam jenis tulisan kaligrafi yang menjadi dataset dengan 60 jumlah data. Nilai akurasi yang diperoleh sebesar 75% [9]. Penelitian menggunakan *Lightweight* CNN untuk mengklasifikasi khat Naskhi dan khat Riq'ah masing-masing menggunakan 100 dataset, dibagi 80:20 untuk data testing, akurasi mencapai 98,75% pada data pelatihan dan 100% pada data validasi [10]. Penelitian analisis perbandingan metode CNN dengan SVM untuk mengklasifikasi gambar jenis tulisan kaligrafi arab. Pada model CNN, diperoleh nilai rata-rata akurasi, presisi, *recall*, dan skor F1 yang terbaik yaitu 94.17%, 94.34%, 94.17%, 94.13%, serta nilai *loss* 0.5840. Sementara pada model SVM menghasilkan rata-rata akurasi, presisi, *recall*, dan skor F1 terbaik yaitu 76.25%, 77.06%, 76.25%, dan 75.87% [5]. Penelitian pengenalan simbol diakritik arab menggunakan CNN arsitektur AlexNet, akurasi keseluruhan mencapai 97% [11]. Penelitian pengenalan karakter khat arab dengan *Multi-Dimensional Long Short-Term Memory* (MDLSTM) dipadukan dengan *Connectionist Temporal Classification* (CTC), sistem mencapai tingkat pengenalan karakter sebesar 80.02% [12]. Penelitian pengenalan teks arab dengan menggabungkan *Convolutional Neural Network* (CNN) dan *Bi-Directional Long Short Term Memory* (BDLSTM), akurasi model mencapai *Character Recognition Rate* (CRR) sebesar 98.76% dan *Word Recognition Rate* (WRR) sebesar 90.22% [13].

Berdasarkan uraian latar belakang dan hasil penelitian terdahulu, dapat disimpulkan bahwa meskipun berbagai metode berbasis *Artificial Intelligence*, khususnya CNN dan model *deep learning* lainnya, telah menunjukkan performa yang baik dalam pengenalan tulisan dan teks Arab, penelitian yang secara khusus membandingkan kinerja CNN dan LVQ pada klasifikasi jenis khat Arab masih terbatas. Selain itu, perbedaan tingkat akurasi pada berbagai penelitian menunjukkan bahwa performa model sangat dipengaruhi oleh arsitektur, teknik ekstraksi fitur, serta karakteristik dataset yang digunakan. Di sisi lain, metode LVQ yang relatif sederhana dan efisien secara komputasi belum banyak dieksplorasi secara komparatif terhadap CNN dalam konteks klasifikasi jenis khat. Oleh karena itu, penelitian ini dilakukan untuk mengisi celah tersebut dengan menganalisis dan membandingkan secara langsung performa CNN dan LVQ dengan menggunakan LBP untuk ekstraksi fitur dalam mengklasifikasikan jenis khat Arab berdasarkan citra digital. Penelitian ini diharapkan dapat memberikan gambaran yang lebih jelas mengenai metode yang paling optimal dalam menangani kompleksitas visual kaligrafi Arab, serta memberikan kontribusi dalam pengembangan sistem klasifikasi khat yang lebih akurat, efisien, dan aplikatif bagi masyarakat maupun peneliti di bidang pengolahan citra dan kecerdasan buatan.

2. METODOLOGI PENELITIAN

Metode penelitian ini disusun untuk menganalisis komparasi metode CNN dan LVQ dalam klasifikasi jenis khat arab, seperti yang ditunjukkan pada Gambar 1.



Gambar 1. Metodologi Penelitian

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 4, June 2026 | Hal 1329-1342
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i4.976

2.1 Pengumpulan Dataset

Dataset yang digunakan dalam penelitian ini diperoleh dari platform *Kaggle.com*, yang terdiri dari lima jenis khat Arab, yaitu Diwani, Farsi, Naskh, Ruqaa, dan Thuluth, dengan total data yang digunakan adalah 640 citra. Dataset dibagi menjadi dua bagian, yaitu data latih sebanyak 600 citra dan data uji sebanyak 40 citra.

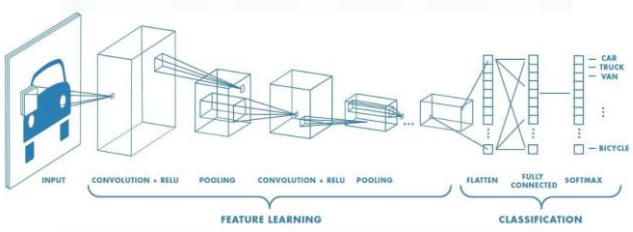
2.2 *Preprocessing*

Digital Image Processing merupakan teknik mengolah citra menggunakan komputer. Citra terdapat beberapa jenis yaitu citra warna (RGB), citra keabu-abuan (*grayscale*), citra biner (monokrom) [14]. *Digital Image Processing* bidang multidisipliner yang bertujuan untuk memperoleh informasi bermakna dari gambar melalui berbagai metode analisis visual [15]. Dengan demikian, proses pra-pemrosesan berperan penting untuk meningkatkan akurasi dan reliabilitas hasil pada tahapan analisis selanjutnya seperti pengenalan pola dan klasifikasi. Tahapan preprocessing meliputi:

a. *Cropping* adalah proses untuk menghilangkan bagian citra yang tidak relevan sehingga fokus utama tertuju pada objek tulisan khat Arab.
b. Konversi ke *Grayscale*, citra RGB dikonversi menjadi citra grayscale untuk menyederhanakan informasi warna tanpa menghilangkan informasi struktur dan tekstur citra.
c. Binerisasi Citra, citra *grayscale* kemudian dikonversi menjadi citra biner dengan tujuan menonjolkan bentuk goresan tulisan khat Arab dan mempermudah proses ekstraksi fitur.

2.3 *Convolutional Neural Network* (CNN)

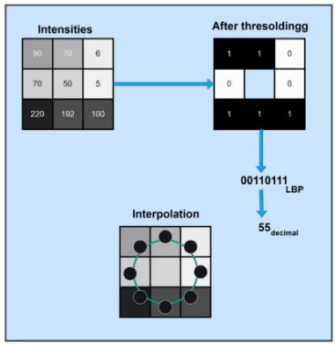
Metode *Convolution Neural Network* (CNN) adalah jenis jaringan saraf tiruan yang dirancang untuk mengolah data berdimensi dua, seperti citra digital. CNN banyak dimanfaatkan dalam tugas deteksi dan pengenalan objek karena kemampuannya mengekstraksi ciri atau fitur penting secara otomatis [16]. Arsitektur CNN memiliki banyak lapisan (*layer*) sehingga kompleksitasnya tinggi dan proses pelatihannya memerlukan waktu yang lama. Untuk mengatasi kendala ini, disarankan penggunaan perangkat komputasi berperforma tinggi. Secara umum, CNN terdiri dari tiga komponen utama, yaitu *Convolution Layer*, *Pooling Layer*, dan *Fully Connected Layer* [17].



Gambar 2. Arsitektur CNN

2.4 *Learning Vector Quantization* (LVQ)

LVQ adalah algoritma klasifikasi untuk mempelajari prototipe yang merepresentasikan wilayah kelas. Wilayah-wilayah tersebut didefinisikan oleh hiperbidang antar prototipe, yang idealnya mendekati batas *Bayesian*. Aturan pembelajaran LVQ modern meminimalkan fungsi biaya eksplisit sehingga mencapai konvergensi yang lebih cepat dan fleksibilitas yang lebih besar. Peningkatan kinerja dalam algoritma LVQ ini telah menyebabkan aplikasinya yang luas di bidang-bidang seperti pemrosesan citra dan sinyal, bioinformatika, dan mekanika [18]. Proses pelatihan model bertujuan untuk menemukan bobot akhir setiap kelas dengan menghitung jarak minimum antara vektor input dan bobot awal menggunakan teknik jarak *Euclidean*. Model klasifikasi akan menghasilkan vektor untuk setiap kelas sebagai representasi dari setiap kelas, yang merupakan bobot akhir setiap fitur. Bobot akhir yang diperoleh akan digunakan dalam proses pengujian [19]. *Local Binary Pattern* (LBP) digunakan untuk ekstraksi fitur pelatihan LVQ. LBP dianggap sebagai salah satu deskriptor tekstur yang paling kuat dan digunakan untuk merepresentasikan fitur lokal suatu citra, yaitu titik-titik penting dalam citra [20].



Gambar 3. *Local Binary Pattern* (LBP)

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 4, June 2026 | Hal 1329-1342
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i4.976

Dalam perhitungan *Local Binary Pattern* (LBP), diberikan nilai "0" ketika nilai piksel pusat lebih besar dari nilai piksel tetangganya, dan nilai "1" ketika lebih kecil. Untuk menentukan nilai LBP piksel pusat, piksel-piksel tetangga dapat ditelusuri searah atau berlawanan arah jarum jam, dimulai dari piksel tetangga mana pun secara bebas. Proses ini menghasilkan bilangan biner 8-bit yang kemudian dapat dikonversi ke bentuk desimal [21].

2.5 Pengujian Model

Model yang telah dilatih kemudian diuji menggunakan 40 data uji. Pengujian dilakukan pada masing-masing model CNN dan LVQ dengan menghitung Akurasi, Presisi, *Recall*, *F1-score.* Selain itu, digunakan *confusion matrix* untuk melihat performa klasifikasi pada setiap kelas khat Arab.

		Actual values	
		+	-
Predicted values	+	True positive(TP)	False positive(FP)
	-	False negative(FN)	True negative (TN)

Gambar 4. *Confusion Matrix*

Berdasarkan nilai-nilai pada *Confusion Matrix*, dimungkinkan untuk menetapkan berbagai ukuran yang memungkinkan kualitas sebuah model dievaluasi. Beberapa ukuran tersebut adalah sebagai berikut [22].

$$Accuracy = \frac{TP + TN}{TP+TN+FP+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

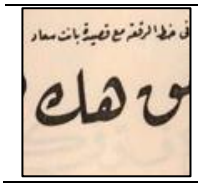
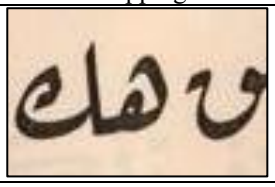
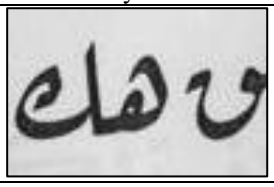
$$F1\ Score = 2\ \frac{P\ x\ R}{P+R} \quad (4)$$

3. HASIL DAN PEMBAHASAN

3.1 *Preprocessing* Dataset

Dataset yang digunakan berjumlah 640 citra khat Arab yang terdiri dari lima kelas, yaitu Diwani, Farsi, Naskh, Ruqaa, dan Thuluth. Setelah dilakukan proses preprocessing berupa *cropping*, konversi ke *grayscale*, dan binerisasi, citra menjadi lebih fokus pada objek tulisan khat dan memiliki kontras yang lebih jelas antara latar belakang dan goresan huruf. Proses *Preprocessing* citra dapat dilihat pada Tabel 1.

Tabel 1. Proses Preprocessing citra

Citra Dataset	Cropping	Grayscale	Binerisasi
			

Hasil preprocessing menunjukkan bahwa proses ini mampu mengurangi noise serta meningkatkan kejelasan pola tulisan, sehingga membantu proses ekstraksi fitur dan pelatihan model baik pada CNN maupun LVQ.

3.2 Pelatihan CNN

Tahapan pertama proses training CNN adalah dengan menentukan parameter training. Parameter pelatihan yang digunakan, yaitu *Learning Rate*, *Epoch*, dan *Batch Size*. Kemudian menentukan arsitektur CNN, seperti pada tabel 2.

Tabel 2. Arsitektur CNN

No	Layer	Konfigurasi	Ukuran Input	Ukuran Output	Jumlah Parameter
1	*Input*	100×100×1	100×100×1	100×100×1	0
2	*Conv2D-1*	Kernel 3×3, 32 filter, *padding same*	100×100×1	100×100×32	320
3	*ReLU*	Aktivasi	100×100×32	100×100×32	0

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 4, June 2026 | Hal 1329-1342
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i4.976

No	Layer	Konfigurasi	Ukuran Input	Ukuran Output	Jumlah Parameter
4	*Max Pooling-1*	2×2, *stride* 2	100×100×32	50×50×32	0
5	*Conv2D-2*	Kernel 3×3, 64 *filter*, *padding same*	50×50×32	50×50×64	18.496
6	*ReLU*	Aktivasi	50×50×64	50×50×64	0
7	*Max Pooling-2*	2×2, *stride* 2	50×50×64	25×25×64	0
8	*Flatten*	-	25×25×64	40.000	0
9	*Fully Connected-1*	64neuron	40.000	64	2.560.064
10	*ReLU*	Aktivasi	64	64	0
11	*Fully Connected Output*	Jumlah Kelas=5	64	5	325
12	*Softmax*	Normalisasi probabilitas	5	5	0
13	*Classification Layer*	*Output* kelas	5	1	0
				Total Parameter	2.579.205

Tahapan selanjutnya menentukan teknik *data augmentation*, dilakukan dengan memutar citra secara acak antara −10° hingga +10° serta menggeser citra secara horizontal dan vertikal masing-masing antara −3 hingga +3 piksel. Proses ini dilakukan secara dinamis saat pelatihan model berlangsung, sehingga setiap epoch menghasilkan variasi citra yang berbeda. *K-Fold Cross Validation* membagi data latih citra dibagi menjadi data latih dan data validasi. Menggunakan nilai K=5. Berikut ini pembagian data dalam Tabel 3..

Tabel 3. Pembagian Data Latih dan Data Validasi

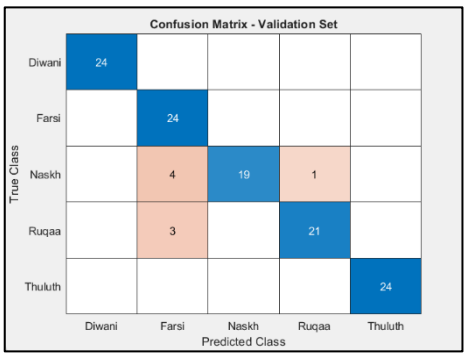
K-Fold	*Cross Validation*				
1	120	120	120	120	120
2	120	120	120	120	120
3	120	120	120	120	120
4	120	120	120	120	120
5	120	120	120	120	120

Pelatihan model CNN ke data latih yang telah dibagi seperti pada Tabel 3. Pada setiap *fold* ditampilkan *confusion matrix* untuk melihat distribusi prediksi benar dan salah pada masing-masing kelas. Dengan pendekatan ini, performa model dapat dianalisis secara menyeluruh berdasarkan hasil validasi di setiap fold. Berikut ini proses pelatihan model CNN pada Gambar 5.

```
===== Fold 1/5 =====
Training on single CPU.
Initializing input data normalization.
|=========================================================================================|
|  Epoch  |  Iteration  |  Time Elapsed  |  Mini-batch  |  Mini-batch  |  Base Learning  |
|         |             |   (hh:mm:ss)   |   Accuracy   |     Loss     |      Rate       |
|=========================================================================================|
|       1 |           1 |       00:00:01 |      34.38% |       1.6013 |          0.0010 |
|       4 |          50 |       00:00:15 |      96.88% |       0.1244 |          0.0010 |
|       7 |         100 |       00:00:29 |      93.75% |       0.2151 |          0.0010 |
|      10 |         150 |       00:00:42 |      93.75% |       0.0975 |          0.0010 |
|      14 |         200 |       00:00:55 |     100.00% |       0.0119 |          0.0010 |
|      17 |         250 |       00:01:12 |     100.00% |       0.0442 |          0.0010 |
|      20 |         300 |       00:01:25 |     100.00% |       0.0173 |          0.0010 |
|      24 |         350 |       00:01:38 |      96.88% |       0.0924 |          0.0010 |
|      27 |         400 |       00:01:51 |     100.00% |       0.0012 |          0.0010 |
|      30 |         450 |       00:02:04 |     100.00% |       0.0013 |          0.0010 |
|=========================================================================================|
Training finished: Max epochs completed.
Accuracy Fold 1 = 93.33 %
```

Gambar 5. Proses Pelatihan Model Fold 1

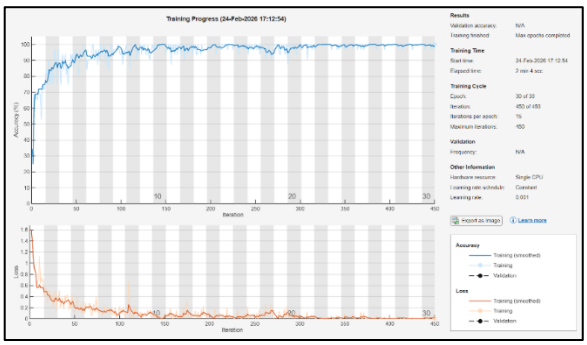
Pada *Fold* 1, Nilai akurasi adalah 93,33% durasi 124 detik. Pelatihan *Fold* 1 dibuktikan dengan confusion matrix pada Gambar 6.



Gambar 6. *Confusion Matrix* Data Validasi *Fold* 1

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 4, June 2026 | Hal 1329-1342
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i4.976

Pada Gambar 6, jumlah data benar adalah 112 data dan 8 data salah dari total 120 data. Dapat dihitung akurasi pelatihan model *Fold* 1 seperti berikut ini. Akurasi *Fold* 1= $\frac{Data\ Benar}{Total\ Data} = \frac{112}{120} = 93{,}3$, Sedangkan grafik pelatihan *Fold* 1 dapat dilihat pada Gambar 7.



Gambar 7. Grafik Pelatihan CNN pada *Fold* 1

Pada Gambar 7 dapat dilihat parameter pelatihan yang digunakan, yaitu *Learning Rate* 0,001, *Epoch* 30, dan *Batch* 32. *Batch* 32 didapatkan dari jumlah data train 480 dibagi jumlah iterasi 15. Pada *Fold* 2 didapatkan akurasi pelatihan yaitu 93,33% dalam waktu 194 detik. Pada *Fold* 3 didapatkan akurasi pelatihan yaitu 95% dalam waktu 213 detik, Pada *Fold* 4 didapatkan akurasi pelatihan yaitu 92,5% dalam waktu 205 detik. Pada *Fold* 5 didapatkan akurasi pelatihan yaitu 92,5% dalam waktu 215 detik. Setelah nilai *Fold* sama dengan K=5, maka dihitunglah nilai rata-rata akurasi, rata-rata waktu pelatihan, dan standar deviasi pelatihan model CNN berdasarkan parameter *Learning Rate* 0,001, *Epoch* 30, dan *Batch* 32, sebagai berikut:

$$Rata-rata\ Akurasi = \frac{93{,}33 + 93{,}33 + 95 + 92{,}5 + 92{,}5}{5} = \frac{466{,}66}{5} = 93{,}33$$

Jadi, Rata-rata akurasi pelatihan yang diperoleh adalah 93,33%

$$Standar\ Deviasi = \sqrt{\frac{(93{,}33-93{,}33)^2 + (93{,}33-93{,}33)^2 + (95-93{,}33)^2 + (92{,}5-93{,}33)^2 + (92{,}5-93{,}33)^2}{5-1}}$$

$$Standar\ Deviasi = \sqrt{\frac{4{,}165}{4}} = \sqrt{1{,}041} \approx 1{,}02$$

Jadi, Standar deviasi pelatihan yang diperoleh adalah ±1,02%

$$Rata-rata\ Durasi = \frac{124 + 194 + 213 + 205 + 215}{5} = \frac{951}{5} = 190{,}20\ detik$$

Jadi, Rata-rata durasi pelatihan yang diperoleh adalah 190,20 detik.

Tahapan selanjutnya adalah melakukan *tuning hyperparameter* untuk mencari kombinasi nilai parameter terbaik pada model agar menghasilkan performa yang paling optimal. *Hyperparameter* ditentukan sebelum training dimulai dan tidak dipelajari otomatis oleh model. Pada Tabel 4 hasil pelatihan CNN yang menggunakan berbagai parameter.

Tabel 4. Hasil *Tuning Hyperparameter* Pelatihan CNN

No	Epoch	Learning Rate	Batch	Akurasi	Standar Deviasi	Durasi (Detik)
1	30	0,01	16	44,83%	34,02%	176,81
2	30	0,01	32	62,83%	39,25%	179,23
3	30	0,01	64	45,67%	35,19%	130,77
4	30	0,005	16	60,00%	36,75%	206,97
5	30	0,005	32	60,00%	36,87%	188,25
6	30	0,005	64	62,50%	38,83%	149,32
7	30	0,001	16	93,83%	2,09%	169,88
8	30	0,001	32	93,33%	1,02%	195,72
9	30	0,001	64	94,00%	1,90%	154,67
10	50	0,01	16	91,25%	1,95%	174,63
11	50	0,01	32	46,33%	36,29%	337,26
12	50	0,01	64	33,50%	30,19%	195,53
13	50	0,005	16	60,50%	37,20%	349,71
14	50	0,005	32	60,89%	37,37%	251,46
15	50	0,005	64	61,41%	36,39%	290,64
16	50	0,001	16	93,83%	1,92%	234,67
17	50	0,001	32	93,83%	2,33%	221,04

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 4, June 2026 | Hal 1329-1342
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i4.976

No	Epoch	Learning Rate	Batch	Akurasi	Standar Deviasi	Durasi (Detik)
18	50	0,001	64	93,83%	3,61%	200,45

Setelah melakukan *tuning hyperparameter*, didapatkan kombinasi parameter terbaik, yaitu akurasi 94%, standar deviasi 1,90% dalam waktu 154,67 detik dengan kombinasi parameter, yaitu *learning rate* 0,001, *epoch* 30, dan *batch* 64. Hasil pelatihan tersebut digunakan sebagai model CNN digunakan pada data uji.

3.3 Pelatihan LVQ

Tahapan proses pelatihan model LVQ dimulai dari menentukan parameter pelatihan, diantaranya *learning rate* awal, *decay rate*, jumlah *epoch*, jumlah prototype per kelas. Kemudian, proses ekstraksi fitur menggunakan *Local Binary Pattern* (LBP) data yang diekstrak berupa citra *grayscale* data latih ukuran 100×100 piksel sebanyak 600, parameter LBP yaitu radius = 1, *neighbors* = 8, *cell size* = [48 48], serta normalisasi menggunakan L2. Hasil ekstraksi fitur dapat dilihat pada Tabel 5.

Tabel 5. Hasil Ekstraksi Fitur Citra oleh *Local Binary Pattern* (LBP)

No	Nama File	Fitur1	Fitur2	Fitur3	…	…	Fitur234	Fitur235	Fitur236
1	Diwani 1.png	0,000011	0,000084	0,000000	…	…	0,007594	0,998614	0,015209
2	Diwani 10.png	0,000494	0,000079	0,000111	…	…	0,002757	0,999330	0,011798
3	Diwani 100.png	0,000298	0,000314	0,000000	…	…	0,002400	0,999023	0,013925
4	Diwani 102.png	0,000000	0,000178	0,000027	…	…	0,006372	0,998260	0,015629
5	Diwani 104.png	0,000475	0,000057	0,000000	…	…	0,003889	0,998859	0,014040
…	…	…	…	…	…	…	…	…	…
…	…	….	…	…	…	…	…	…	…
596	Thuluth 95.jpg	0,002067	0,000667	0,000000	…	…	0,003783	0,999804	0,010979
597	Thuluth 96.jpg	0,001999	0,000452	0,000000	…	…	0,000370	0,999993	0,002008
598	Thuluth 97.jpg	0,001351	0,001424	0,000070	…	…	0,003171	0,999729	0,013938
599	Thuluth 98.jpg	0,001671	0,001023	0,000000	…	…	0,001881	0,999897	0,007415
600	Thuluth 99.jpg	0,001626	0,001021	0,000000	…	…	0,003519	0,999669	0,015202

Pada Tabel 5 dapat dilihat hasil ekstraksi fitur citra oleh LBP berupa numerik dan menghasilkan 236 fitur. Selanjutnya menyimpan hasil ekstraksi fitur dalam bentuk matriks. Tujuannya agar data lebih terstruktur dan siap diproses pada tahap pelatihan model. Dalam matriks ini, setiap baris mewakili satu citra, sedangkan setiap kolom merupakan nilai dari fitur tertentu. Kemudian normalisasi fitur dengan L2, Tujuan melakukan normalisasi fitur adalah untuk menyamakan skala nilai antar fitur supaya tidak ada fitur yang mendominasi proses pembelajaran model. Saat ekstraksi LBP, setiap citra menghasilkan ratusan atau ribuan nilai fitur. Nilai tersebut bisa punya rentang yang berbeda-beda. Dengan adanya normalisasi fitur, nilai ekstraksi siap untuk diproses algoritma LVQ. Tabel 6 menampilkan tabel hasil normalisasi.

Tabel 6. Hasil Normalisasi Fitur

No	Nama File	Fitur1	Fitur2	Fitur3	…	…	Fitur234	Fitur235	Fitur236
1	Diwani 1.png	-0,5979	-0,57765	-0,50426	…	…	-0,33941	0,524763	-0,80948
2	Diwani 10.png	-0,58919	-0,57782	-0,41748	…	…	-0,58933	0,53013	-0,82753
3	Diwani 100.png	-0,59272	-0,56982	-0,50426	…	…	-0,60781	0,527825	-0,81628
4	Diwani 102.png	-0,59809	-0,57444	-0,48358	…	…	-0,40257	0,522112	-0,80726
5	Diwani 104.png	-0,58952	-0,57857	-0,50426	…	…	-0,53085	0,526602	-0,81567
…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…
596	Thuluth 95.jpg	-0,56083	-0,55778	-0,50426	…	…	-0,53634	0,533678	-0,83186
597	Thuluth 96.jpg	-0,56205	-0,56511	-0,50426	…	…	-0,71271	0,535094	-0,87933
598	Thuluth 97.jpg	-0,57373	-0,53195	-0,44959	…	…	-0,56797	0,533121	-0,81621
599	Thuluth 98.jpg	-0,56796	-0,54561	-0,50426	…	…	-0,63464	0,534375	-0,85072
600	Thuluth 99.jpg	-0,56877	-0,54568	-0,50426	…	…	-0,54996	0,532664	-0,80952

Setelah ekstraksi fitur, data tersebut akan dibagi oleh *K-Fold Cross Validation* menjadi data latih dan data validasi. Dengan nilai K=5, dari total data pelatihan 600 citra maka jumlah data untuk pelatihan model adalah 480 dan sisanya sebanyak 120 menjadi data validasi. Hasil pembagiannya dapat dilihat pada Tabel 3 sebelumnya.

Dalam pelatihan ini jumlah prototipe awal yang digunakan adalah 5 prototipe perkelas, jadi total ada 25 prototipe awal.Tabel 7 menampilkan data prototipe awal yang akan digunakan untuk pelatihan model LVQ.

Tabel 7. Fitur Prototipe Awal Pelatihan LVQ

	Nama File	Fitur 1	Fitur 2	Fitur 3	…	Fitur 234	Fitur 235	Fitur 236	Label Kelas
W_1	Diwani 109.png	-0,591893	-0,580522	-0,464081	…	-0,528376	0,525138	-0,808765	Diwani
W_2	Diwani 20.png	-0,592215	-0,570633	-0,504260	…	-0,528903	0,528261	-0,822850	Diwani
W_3	Diwani 64.png	-0,593535	-0,578773	-0,504260	…	-0,536301	0,527896	-0,832116	Diwani

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 4, June 2026 | Hal 1329-1342
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i4.976

	Nama File	Fitur 1	Fitur 2	Fitur 3	…	Fitur 234	Fitur 235	Fitur 236	Label Kelas
W_4	Diwani 79.png	-0,596542	-0,578277	-0,504260	…	-0,514504	0,528555	-0,806413	Diwani
W_5	Diwani 92.png	-0,594700	-0,578012	-0,504260	…	-0,508890	0,520820	-0,788935	Diwani
W_6	Farsi 108.png	-0,079962	-0,089534	-0,031163	…	-0,425262	0,489927	-0,395821	Farsi
W_7	Farsi 16.png	-0,326255	-0,173906	0,413863	…	-0,454064	0,445789	-0,167082	Farsi
W_8	Farsi 44.png	-0,121884	0,141993	-0,151560	…	-0,233675	0,309016	0,233434	Farsi
W_9	Farsi 71.png	0,575094	0,331244	0,861366	…	-0,283876	0,395182	0,029358	Farsi
W_{10}	Farsi 93.png	-0,345117	-0,256964	-0,354619	…	-0,137943	0,160869	0,581325	Farsi
W_{11}	Naskh 100.png	-0,594828	-0,575117	-0,360148	…	-0,514280	0,514108	-0,771792	Naskh
W_{12}	Naskh 15.png	2,070630	3,569753	0,889201	…	1,353110	-1,860474	1,559839	Naskh
W_{13}	Naskh 53.png	2,383748	1,345144	1,704569	…	1,066657	-1,547784	2,061845	Naskh
W_{14}	Naskh 68.png	2,770417	0,609026	3,502784	…	3,229339	-3,167736	2,848818	Naskh
W_{15}	Naskh 84.png	-0,596327	-0,566539	-0,504260	…	-0,562323	0,525407	-0,824056	Naskh
W_{16}	Ruqaa 106.png	-0,072369	-0,062545	-0,137251	…	-0,177082	0,204368	0,500704	Ruqaa
W_{17}	Ruqaa 126.png	3,110546	0,772345	6,311012	…	3,601836	-3,137632	2,202750	Ruqaa
W_{18}	Ruqaa 24.png	-0,360348	-0,332601	-0,206214	…	-0,043394	0,403296	-0,017232	Ruqaa
W_{19}	Ruqaa 39.png	0,272063	0,210696	-0,346207	…	-0,189753	0,369206	0,086328	Ruqaa
W_{20}	Ruqaa 60.png	-0,087759	-0,066133	-0,212791	…	-0,304577	0,241711	0,411812	Ruqaa
W_{21}	Thuluth 102.png	-0,593541	-0,580522	-0,485694	…	-0,606233	0,533640	-0,836105	Thuluth
W_{22}	Thuluth 12.png	-0,569241	-0,565725	-0,504260	…	-0,644341	0,534438	-0,855879	Thuluth
W_{23}	Thuluth 24.png	-0,576687	-0,580522	-0,243173	…	-0,375319	0,531011	-0,818912	Thuluth
W_{24}	Thuluth 42.png	-0,575858	-0,544461	-0,504260	…	-0,477793	0,530711	-0,804524	Thuluth
W_{25}	Thuluth 76.png	-0,571518	-0,547445	-0,504260	…	-0,603743	0,533689	-0,840603	Thuluth

Sesudah menentukan prototipe awal, tahapan selanjutnya adalah melakukan proses data sampel pertama (x_1). Contoh data sampel pertama yang akan diproses adalah data Diwani 40.png dengan fitur {-0,596836, -0,578808, -0,504260, …, -0,582936, 0,531890, -0,849400}. Dengan *Learning Rate* awal η=0.1. Dapat dihitung jarak *Euclidean* ke fitur prototipe W_1 sampai dengan W_{25}:

Jarak ke W_1

$$d_1 = \sqrt{(-0{,}59683 - (-0{,}591893))^2 + \left(-0{,}578808 - (-0{,}580522)\right)^2 + \cdots + (0{,}531890 - 0{,}525138)^2 + (-0{,}849400 - (-0{,}808765))^2} = 1{,}105$$

Jarak ke W_2

$$d_2 = \sqrt{(-0{,}59683 - (-0{,}592215))^2 + \left(-0{,}578808 - (-0{,}570633)\right)^2 + \cdots + (0{,}531890 - 0{,}528261)^2 + (-0{,}849400 - (-0{,}822850))^2} = 0{,}792$$

Jarak ke W_3

$$d_3 = \sqrt{(-0{,}59683 - (-0{,}593535))^2 + \left(-0{,}578808 - (-0{,}578773)\right)^2 + \cdots + (0{,}531890 - 0{,}527896)^2 + (-0{,}849400 - (-0{,}832116))^2} = 1{,}140$$

Dan seterusnya sampai

Jarak ke W_{23}

$$d_{23} = \sqrt{(-0{,}59683 - (-0{,}576687))^2 + \left(-0{,}578808 - (-0{,}580522)\right)^2 + \cdots + (0{,}531890 - 0{,}531011)^2 + (-0{,}849400 - (-0{,}818912))^2} = 1{,}811$$

Jarak ke W_{24}

$$d_{24} = \sqrt{(-0{,}59683 - (-0{,}575858))^2 + \left(-0{,}578808 - (-0{,}544461)\right)^2 + \cdots + (0{,}531890 - 0{,}530711)^2 + (-0{,}849400 - (-0{,}804524))^2} = 1{,}043$$

Jarak ke W_{25}

$$d_{25} = \sqrt{(-0{,}59683 - (-0{,}571518))^2 + \left(-0{,}578808 - (-0{,}547445)\right)^2 + \cdots + (0{,}531890 - 0{,}533689)^2 + (-0{,}849400 - (-0{,}8840603))^2} = 1{,}433$$

Setelah jarak *Euclidean* masing-masing prototipe didapatkan selanjutnya menentukan *winner*, *winner* adalah prototipe yang mempunyai nilai *Euclidean* terkecil. Disini yang mendekati adalah nilai d_4 yaitu label Diwani. Setelah *winner* didapatkan yaitu W_4, maka dilakukan update bobot prototipe W_4 dengan hitungan:

$$W_{4,1}^{new} = -0{,}596542 + 0{,}1 \times \left(-0{,}596836 - (-0{,}596542)\right) = -0{,}596571$$

$$W_{4,2}^{new} = -0{,}578277 + 0{,}1 \times \left(-0{,}578808 - (-0{,}578277)\right) = -0{,}578330$$

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 4, June 2026 | Hal 1329-1342
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i4.976

Dan seterusnya sampai

$$W_{4,235}^{new} = 0{,}528555 + 0{,}1 \times (0{,}531890 - 0{,}528555) = 0{,}528888$$

$$W_{4,236}^{new} = -0{,}806413 + 0{,}1 \times \left(-0{,}849400 - (-0{,}806413)\right) = -0{,}810711$$

Jadi bobot W_4^{new} menjadi:

$$W_4^{new} = [-0{,}596571, -0{,}578330, \ldots, 0{,}528888, -0{,}810711]$$

Kemudian hitung jarak baru x_1 ke W_4^{new}

$$d_4 = \sqrt{(-0{,}59683 - (-0{,}596571))^2 + \left(-0{,}578808 - (-0{,}578330)\right)^2 + \cdots + (0{,}531890 - 0{,}528888)^2 + (-0{,}849400 - (-0{,}810711))^2} = 0{,}486$$

Perbandingan: jarak awal $d_4 = 0.593$, setelah update $d_4^{new} = 0.486$ jadi berkurang, menunjukkan bahwa prototipe mengalami pembaruan mendekati data sampel. Hal ini mengindikasikan bahwa label prototipe sesuai dengan label data, sehingga dilakukan proses koreksi dengan mendorong prototipe mendekati untuk memperjelas batas antar kelas. Proses pembaruan bobot ini bertujuan untuk memperjelas batas antar kelas agar model dapat membedakan pola fitur dengan lebih baik. Setelah pembaruan dilakukan, proses dilanjutkan ke data latih berikutnya hingga seluruh data dalam satu siklus selesai diproses. Satu kali pemrosesan seluruh data latih disebut sebagai satu *epoch*. Pada akhir setiap *epoch*, dapat dihitung nilai *error* atau akurasi data latih untuk melihat perkembangan pembelajaran. *Learning rate* kemudian diturunkan secara bertahap sebesar *decay rate* agar proses pembelajaran menjadi lebih stabil pada iterasi selanjutnya. Tahapan ini diulang hingga mencapai jumlah *epoch* maksimum atau 1 *fold.* Gambar 8 menunjukkan hasil proses pelatihan LVQ pada *Fold* 1.

```
===== Fold 1/5 =====
Epoch 01 | Error = 0.1646 | Acc Train = 88.75% | eta = 0.05000
Epoch 02 | Error = 0.1250 | Acc Train = 88.75% | eta = 0.04500
Epoch 03 | Error = 0.1229 | Acc Train = 88.75% | eta = 0.04050
Epoch 04 | Error = 0.1146 | Acc Train = 88.75% | eta = 0.03645
Epoch 05 | Error = 0.1104 | Acc Train = 90.21% | eta = 0.03281
Epoch 06 | Error = 0.1062 | Acc Train = 89.79% | eta = 0.02952
Epoch 07 | Error = 0.1083 | Acc Train = 90.21% | eta = 0.02657
Epoch 08 | Error = 0.1000 | Acc Train = 89.79% | eta = 0.02391
Epoch 09 | Error = 0.0958 | Acc Train = 90.21% | eta = 0.02152
Epoch 10 | Error = 0.1021 | Acc Train = 90.62% | eta = 0.01937
Epoch 11 | Error = 0.0958 | Acc Train = 90.42% | eta = 0.01743
Epoch 12 | Error = 0.0958 | Acc Train = 90.83% | eta = 0.01569
Epoch 13 | Error = 0.0979 | Acc Train = 90.42% | eta = 0.01412
Epoch 14 | Error = 0.1000 | Acc Train = 90.62% | eta = 0.01271
Epoch 15 | Error = 0.0958 | Acc Train = 90.62% | eta = 0.01144
Epoch 16 | Error = 0.0938 | Acc Train = 90.62% | eta = 0.01029
Epoch 17 | Error = 0.0917 | Acc Train = 90.62% | eta = 0.00927
Epoch 18 | Error = 0.0958 | Acc Train = 90.62% | eta = 0.00834
Epoch 19 | Error = 0.0958 | Acc Train = 90.42% | eta = 0.00750
Epoch 20 | Error = 0.0979 | Acc Train = 90.62% | eta = 0.00675
Epoch 21 | Error = 0.0938 | Acc Train = 90.62% | eta = 0.00608
Epoch 22 | Error = 0.0958 | Acc Train = 90.62% | eta = 0.00547
Epoch 23 | Error = 0.0958 | Acc Train = 90.62% | eta = 0.00492
Epoch 24 | Error = 0.0958 | Acc Train = 90.62% | eta = 0.00443
Epoch 25 | Error = 0.0958 | Acc Train = 90.62% | eta = 0.00399
Epoch 26 | Error = 0.0979 | Acc Train = 90.62% | eta = 0.00359
Epoch 27 | Error = 0.0979 | Acc Train = 90.42% | eta = 0.00323
Epoch 28 | Error = 0.0958 | Acc Train = 90.42% | eta = 0.00291
Epoch 29 | Error = 0.0938 | Acc Train = 90.42% | eta = 0.00262
Epoch 30 | Error = 0.0958 | Acc Train = 90.42% | eta = 0.00236
Accuracy Fold 1 = 85.83 %
Duration Fold 1 = 0.30 detik
```

Gambar 8. Proses Pelatihan LVQ pada *Fold* 1

Berdasarkan Gambar 8, akurasi pelatihan pada Fold 1 yaitu 85% dalam waktu 0,3 detik. Pada *Fold* 2 akurasi pelatihan yaitu 87,5% dalam waktu 0,28 detik. Pada *Fold* 3 akurasi pelatihan yaitu 89,17% dalam waktu 0,29 detik. Pada *Fold* 4 akurasi pelatihan yaitu 90,83% dalam waktu 0,29 detik. Pada *Fold* 5 akurasi pelatihan yaitu 91,67% dalam waktu 0,29 detik. Setelah nilai *Fold* sama dengan nilai K=5, maka dihitunglah nilai rata-rata akurasi, rata-rata waktu pelatihan, dan standar deviasi pelatihan model LVQ berdasarkan parameter *Learning Rate* awal 0,05, *Epoch* 30, dan *Decay rate* 0,1, sebagai berikut:

$$Rata - rata\ Akurasi = \frac{85{,}83 + 87{,}50 + 89{,}17 + 90{,}83 + 91{,}67}{5} = \frac{445}{5} = 89$$

Jadi, Rata-rata akurasi pelatihan yang diperoleh adalah 89%

$$Standar\ Deviasi = \sqrt{\frac{(85{,}83 - 89)^2 + (87{,}5 - 89)^2 + (89{,}17 - 89)^2 + (90{,}83 - 89)^2 + (91{,}67 - 89)^2}{5 - 1}}$$

$$Standar\ Deviasi = \sqrt{\frac{22{,}81}{4}} = \sqrt{5{,}7025} \approx 2{,}39$$

Jadi, Standar deviasi pelatihan yang diperoleh adalah ±2,39%

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 4, June 2026 | Hal 1329-1342
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i4.976

$$Rata-rata\ Durasi = \frac{0{,}30 + \ 0{,}28 + 0{,}29 + 0{,}29 + \ 0{,}29}{5} = \frac{1{,}45}{5} = 0{,}29\ detik$$

Jadi, Rata-rata durasi pelatihan yang diperoleh adalah 0,29 detik.

Pada tahapan *tuning hyperparameter*, hasil pelatihan model LVQ dapat dilihat pada Tabel 8.

Tabel 8. Hasil *Tuning Hyperparameter* Pelatihan LVQ

No	Epoch	Learning Rate Awal	Decay Rate	Akurasi	Standar Deviasi	Durasi
1	30	0,01	0,05	86,50%	5,73%	0,24
2	30	0,01	0,1	87,67%	3,03%	0,23
3	30	0,01	0,5	83,17%	2,97%	0,23
4	30	0,05	0,05	87,50%	2,83%	0,24
5	30	0,05	0,1	89,00%	2,39%	0,29
6	30	0,05	0,5	84,00%	3,41%	0,24
7	30	0,1	0,05	91,33%	2,09%	0,24
8	30	0,1	0,1	88,50%	3,30%	0,24
9	30	0,1	0,5	87,67%	2,46%	0,24
10	50	0,01	0,05	85,33%	2,67%	0,40
11	50	0,01	0,1	87,50%	2,50%	0,40
12	50	0,01	0,5	82,83%	1,73%	0,40
13	50	0,05	0,05	86,83%	2,73%	0,40
14	50	0,05	0,1	88,17%	4,84%	0,39
15	50	0,05	0,5	86,67%	2,64%	0,40
16	50	0,1	0,05	86,50%	4,01%	0,40
17	50	0,1	0,1	90,67%	2,53%	0,44
18	50	0,1	0,5	87,50%	2,04%	0,51

Setelah melakukan *tuning hyperparameter*, didapatkan kombinasi parameter terbaik, yaitu akurasi 91,33%, standar deviasi 2,09% dalam waktu 0,24 detik dengan kombinasi parameter, yaitu *learning rate* awal 0,001, *epoch* 30, dan *decay rate* 0,05. Hasil pelatihan tersebut digunakan sebagai model LVQ digunakan pada data uji.

3.4 Pengujian CNN

Setelah Model CNN didapatkan, kemudian dilakukan pengujian model memakai data uji. Data uji yang digunakan adalah sebesar 40 data citra dengan 8 citra masing-masing kelas khat arab. Hasil Pengujian dapat dilihat pada Tabel 9.

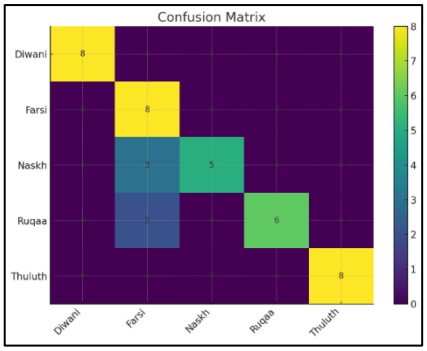
Tabel 9. Hasil Pengujian Model CNN

No	Data Uji	Hasil Prediksi	Berhasil/Gagal
1	Diwani	Diwani	Berhasil
2	Diwani	Diwani	Berhasil
3	Diwani	Diwani	Berhasil
4	Diwani	Diwani	Berhasil
5	Diwani	Diwani	Berhasil
6	Diwani	Diwani	Berhasil
7	Diwani	Diwani	Berhasil
8	Diwani	Diwani	Berhasil
9	Farsi	Farsi	Berhasil
10	Farsi	Farsi	Berhasil
11	Farsi	Farsi	Berhasil
12	Farsi	Farsi	Berhasil
13	Farsi	Farsi	Berhasil
14	Farsi	Farsi	Berhasil
15	Farsi	Farsi	Berhasil
16	Farsi	Farsi	Berhasil
17	Naskh	Farsi	Gagal
18	Naskh	Naskh	Berhasil
19	Naskh	Naskh	Berhasil
20	Naskh	Farsi	Gagal
21	Naskh	Naskh	Berhasil
22	Naskh	Naskh	Berhasil
23	Naskh	Farsi	Gagal
24	Naskh	Naskh	Berhasil
25	Ruqaa	Ruqaa	Berhasil
26	Ruqaa	Farsi	Gagal

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 4, June 2026 | Hal 1329-1342
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i4.976

No	Data Uji	Hasil Prediksi	Berhasil/Gagal
27	Ruqaa	Ruqaa	Berhasil
28	Ruqaa	Ruqaa	Berhasil
29	Ruqaa	Farsi	Gagal
30	Ruqaa	Ruqaa	Berhasil
31	Ruqaa	Ruqaa	Berhasil
32	Ruqaa	Ruqaa	Berhasil
33	Thuluth	Thuluth	Berhasil
34	Thuluth	Thuluth	Berhasil
35	Thuluth	Thuluth	Berhasil
36	Thuluth	Thuluth	Berhasil
37	Thuluth	Thuluth	Berhasil
38	Thuluth	Thuluth	Berhasil
39	Thuluth	Thuluth	Berhasil
40	Thuluth	Thuluth	Berhasil

Berdasarkan data pada Tabel 9 terdapat 35 data yang diprediksi benar dan 5 data yang diprediksi salah. *Confusion Matrix* Model CNN dapat dilihat pada Gambar 9.



Gambar 9. *Confusion Matrix* Pengujian Model CNN

Untuk mengukur evaluasi kinerja model CNN dapat diukur dengan menghitung *Accuracy, Precision, Recall* dan *F1-Score* berdasarkan *confusion matrix* yang telah ada.

$$Accuracy = \frac{8+5+8+5+8}{8+8+8+8+8} = \frac{34}{40} = 0{,}875$$

Accuracy CNN = 87,5%

Kelas Diwani (TP = 8, FP = 0, FN = 0)

$$Precision = \frac{8}{8+0} = 1{,}00, \quad Recall = \frac{8}{8+0} = 1{,}00, \quad F1 = 2x\frac{1\ x\ 1}{1+1} = 1{,}00$$

Kelas Farsi (TP = 8, FP = 5, FN = 0)

$$Precision = \frac{8}{8+5} = 0{,}615, \quad Recall = \frac{8}{8+0} = 1{,}00, \quad F1 = 2x\frac{0{,}615\ x\ 1}{0{,}615+1} = 0{,}762$$

Kelas Naskh (TP = 5, FP = 0, FN = 3)

$$Precision = \frac{5}{5+0} = 1{,}00, \quad Recall = \frac{5}{5+3} = 0{,}625, \quad F1 = 2x\frac{1\ x\ 0{,}625}{1+0{,}625} = 0{,}769$$

Kelas Ruqaa (TP = 6, FP = 0, FN = 2)

$$Precision = \frac{6}{6+0} = 1{,}00, \quad Recall = \frac{6}{6+2} = 0{,}75, \quad F1 = 2x\frac{1\ x\ 0{,}75}{1+0{,}75} = 0{,}857$$

Kelas Thuluth (TP = 8, FP = 0, FN = 0)

$$Precision = \frac{8}{8+0} = 1{,}00, \quad Recall = \frac{8}{8+0} = 1{,}00, \quad F1 = 2x\frac{1\ x\ 1}{1+1} = 1{,}00$$

Ringkasan Evaluasi kinerja Model CNN dapat dilihat pada Tabel 10

Tabel 10. Evaluasi Kinerja CNN

Kelas	Precision	Recall	F1
Diwani	1,00	1,00	1,00

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 4, June 2026 | Hal 1329-1342
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i4.976

Kelas	Precision	Recall	F1
Farsi	0,615	1,00	0,762
Naskh	1,00	0,625	0,769
Ruqaa	1,00	0,75	0,857
Thuluth	1,00	1,00	1,00
Macro Precision		0,923	
Macro Recall		0,878	
Macro F1 Score		0,878	
Accuracy		87,5%	

3.5 Pengujian LVQ

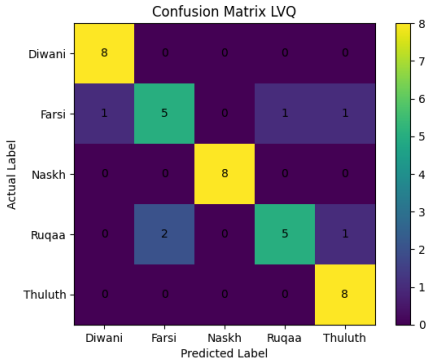
Pada model LVQ yang telah dilatih, pengujian model dengan data uji menggunakan 40 data bisa dilihat pada Tabel 11.

Tabel 11. Hasil Pengujian Model LVQ

No	Data Uji	Hasil Prediksi	Berhasil/Gagal
1	Diwani	Diwani	Berhasil
2	Diwani	Diwani	Berhasil
3	Diwani	Diwani	Berhasil
4	Diwani	Diwani	Berhasil
5	Diwani	Diwani	Berhasil
6	Diwani	Diwani	Berhasil
7	Diwani	Diwani	Berhasil
8	Diwani	Diwani	Berhasil
9	Farsi	Farsi	Berhasil
10	Farsi	Farsi	Berhasil
11	Farsi	Diwani	Gagal
12	Farsi	Ruqaa	Gagal
13	Farsi	Farsi	Berhasil
14	Farsi	Farsi	Berhasil
15	Farsi	Thuluth	Gagal
16	Farsi	Farsi	Berhasil
17	Naskh	Naskh	Berhasil
18	Naskh	Naskh	Berhasil
19	Naskh	Naskh	Berhasil
20	Naskh	Naskh	Berhasil
21	Naskh	Naskh	Berhasil
22	Naskh	Naskh	Berhasil
23	Naskh	Naskh	Berhasil
24	Naskh	Naskh	Berhasil
25	Ruqaa	Farsi	Gagal
26	Ruqaa	Farsi	Gagal
27	Ruqaa	Ruqaa	Berhasil
28	Ruqaa	Ruqaa	Berhasil
29	Ruqaa	Ruqaa	Berhasil
30	Ruqaa	Ruqaa	Berhasil
31	Ruqaa	Thuluth	Gagal
32	Ruqaa	Ruqaa	Berhasil
33	Thuluth	Thuluth	Berhasil
34	Thuluth	Thuluth	Berhasil
35	Thuluth	Thuluth	Berhasil
36	Thuluth	Thuluth	Berhasil
37	Thuluth	Thuluth	Berhasil
38	Thuluth	Thuluth	Berhasil
39	Thuluth	Thuluth	Berhasil
40	Thuluth	Thuluth	Berhasil

Berdasarkan data pada tabel 11 terdapat 34 data yang diprediksi benar dan 6 data yang diprediksi salah. *Confusion matrix* pengujian Model LVQ dapat dilihat pada gambar 10.

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 4, June 2026 | Hal 1329-1342
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i4.976



Gambar 10. *Confusion Matrix* Pengujian Model LVQ

Sedangkan hasil *confusion matrix* LVQ adalah sebagai berikut:

$$Accuracy = \frac{8+5+8+5+8}{8+8+8+8+8} = \frac{34}{40} = 0{,}85$$

Accuracy LVQ = 85%

Kelas Diwani (TP = 8, FP = 1, FN = 0)

$$Precision = \frac{8}{8+1} = 0{,}889, \quad Recall = \frac{8}{8+0} = 1{,}00, \quad F1 = 2x\frac{0{,}889\ x\ 1}{0{,}889+1} = 0{,}941$$

Kelas Farsi (TP = 5 FP = 2, FN = 3)

$$Precision = \frac{5}{5+2} = 0{,}714, \quad Recall = \frac{5}{5+3} = 0{,}625, \quad F1 = 2x\frac{0{,}714\ x\ 0{,}625}{0{,}714+0{,}625} = 0{,}667$$

Kelas Naskh (TP = 8, FP = 0, FN = 0)

$$Precision = \frac{8}{8+0} = 1{,}00, \quad Recall = \frac{8}{8+0} = 1{,}00, \quad F1 = 2x\frac{1{,}00\ x\ 1{,}00}{1{,}00+1{,}00} = 1{,}00$$

Kelas Ruqaa (TP = 5, FP = 1, FN = 3)

$$Precision = \frac{5}{5+1} = 0{,}833, \quad Recall = \frac{5}{5+3} = 0{,}625, \quad F1 = 2x\frac{0{,}833\ x\ 0{,}625}{0{,}833+0{,}625} = 0{,}714$$

Kelas Thuluth (TP = 8, FP = 2, FN = 0)

$$Precision = \frac{8}{8+2} = 0{,}80, \quad Recall = \frac{8}{8+0} = 1{,}00, \quad F1 = 2x\frac{0{,}8\ x\ 1}{0{,}8+1} = 0{,}889$$

Ringkasan Evaluasi kinerja Model LVQ dapat dilihat pada Tabel 12.

Tabel 12. Evaluasi Kinerja LVQ

Kelas	Precision	Recall	F1
Diwani	0,889	1,000	0,941
Farsi	0,714	0,625	0,667
Naskh	1,000	1,000	1,000
Ruqaa	0,833	0,625	0,714
Thuluth	0,800	1,000	0,889
Macro Precision		0,847	
Macro Recall		0,850	
Macro F1-Score		0,842	
Accuracy		85%	

4. KESIMPULAN

Berdasarkan hasil evaluasi kinerja pada Tabel 10 dan Tabel 12, dilakukan perbandingan antara metode *Convolutional Neural Network* (CNN) dan *Learning Vector Quantization* (LVQ) dalam mengklasifikasikan lima jenis khat Arab, yaitu Diwani, Farsi, Naskh, Ruqaa, dan Thuluth. Evaluasi menggunakan metrik *precision, recall, F1-score*, serta *accuracy* untuk melihat performa model secara menyeluruh maupun per kelas. Secara umum CNN menunjukkan performa yang sedikit lebih baik dibandingkan LVQ. Model CNN memperoleh nilai *accuracy* sebesar 87,5%, sedangkan LVQ memperoleh 85%. Dari sisi rata-rata metrik (*macro average*), CNN juga memiliki nilai yang lebih tinggi, yaitu *macro precision* sebesar 0,923, *macro recall* sebesar 0,878, dan *macro F1-score* sebesar 0,878. Sementara itu, LVQ memperoleh

macro precision sebesar 0,847, *macro recall* sebesar 0,850, dan *macro F1-score* sebesar 0,842. Hal ini menunjukkan bahwa secara keseluruhan CNN lebih konsisten dalam mengklasifikasikan seluruh kelas dan juga lebih unggul dalam menangkap kompleksitas visual citra digital khat Arab, sedangkan LVQ dengan dibantu LBP untuk ekstraksi fitur menawarkan performa kompetitif dengan arsitektur yang lebih sederhana dan efisiensi komputasi yang lebih cepat dibandingkan CNN. Dengan demikian, metode CNN lebih direkomendasikan untuk digunakan dalam sistem klasifikasi citra khat Arab karena memiliki kemampuan yang lebih baik dalam mengekstraksi dan mengenali karakteristik visual citra digital khat arab yang kompleks.

REFERENCES

[1] A. Muiz, "Peran Khat Kaligrafi dalam Meningkatkan Estetika dan Pemahaman," *Asian Journal Of Multidisciplinary Research Asian Journal Of Multidisciplinary Research*," vol. 1, no. 2, 2024, url: https://jujurnal.com/index.php/ajmr/article/download/55/43

[2] S. Soaleha, N. Amin, and A. B. Malla, "Kemampuan Menulis Kaligrafi Mahasiswa Sastra Arab Universitas Muslim Indonesia Makassar," *Karya Ilmiah Mahasiswa-KIMA*, vol. 1, no. 2, 2022, url: https://jurnal.fs.umi.ac.id/index.php/KIMA/article/view/452

[3] D. Bhatt *et al.*, "Cnn variants for computer vision: History, architecture, application, challenges and future scope," *electronics*, Oct. 01, 2021, *MDPI*. doi: 10.3390/electronics10202470.

[4] L. Chen, S. Li, Q. Bai, J. Yang, S. Jiang, and Y. Miao, "Review of image classification algorithms based on convolutional neural networks," Nov. 01, 2021, *MDPI*. doi: 10.3390/rs13224712.

[5] Afriani, H. Sujaini, N. Candraningrum, J. H. Hadari Nawawi "Perbandingan Metode Pengklasifikasi Gambar Jenis Tulisan Kaligrafi Arab, "*JEPIN (Jurnal Edukasi dan Penelitian Informatika)*," vol. 10, no. 1, pp. 68-78, 2024, doi: 10.26418/jp.v10i1.72863

[6] Purwono, A. Ma'arif, W. Rahmaniar, H. I. K. Fathurrahman, A. Z. K. Frisky, and Q. M. U. Haq, "Understanding of Convolutional Neural Network (CNN): A Review," *International Journal of Robotics and Control Systems*, vol. 2, no. 4, pp. 739–748, 2022, doi: 10.31763/ijrcs.v2i4.888.

[7] R. Van Veen, M. Biehl, and M. B. Nl, "sklvq: Scikit Learning Vector Quantization," 2021. [Online]. Available: https://sklvq.readthedocs.io/en/0.1.2/

[8] N. Kasim and G. Satya Nugraha, "Pengenalan Pola Tulisan Tangan Aksara Arab Menggunakan Metode Convolution Neural Network (Handwritten Arabic Script Recognition Using Convolution Neural Network)." *Jurnal Teknologi Informasi Komputer, (jtika)*, vol. 3, No. 1, pp. 85-95, 2021, doi: 10.29303/jtika.v3i1.136

[9] M. Furqan, A. H. Hasugian, Z. Addilah, and G. Artikel, "Pengenalan Jenis Teks Kaligrafi Menggunakan Learning Vector Quantization Calligraphy Text Types Recognition Using Learning Vector Quantization Article Info ABSTRAK," *JOMLAI: Journal of Machine Learning and Artificial Intelligence*, vol. 1, no. 4, pp. 2828–9099, 2022, doi: 10.55123/jomlai.v1i4.1653.

[10] J. Mantik, M. Taufiq Riza, and O. Virgantara Putra, "Lightweight convolutional neural network for khat naskhi and riq'ah classification," Online, 2023.

[11] Y. Alalawi, D. M. Chandler, and N. R. Caluya, "A CNN-Based Arabic Diacritic Symbol Recognition System Using Domain Adaptation," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Oct. 2023, pp. 23–32. doi: 10.1145/3626641.3627212.

[12] R. Ahmad, S. Naz, M. Afzal, S. Rashid, M. Liwicki, and A. Dengel, "A deep learning based arabic script recognition system: Benchmark on khat," *International Arab Journal of Information Technology*, vol. 17, no. 3, pp. 299–305, May 2020, doi: 10.34028/iajit/17/3/3.

[13] M. Fasha, J. Bassam Hammo, N. Obeid, and J. AlWidian, "A Hybrid Deep Learning Model for Arabic Text Recognition," 2020. [Online]. Available: www.ijacsa.thesai.org

[14] A. Fadjeri, L. Kurniatin, D. K. Adri Ariyanto, and B. A. Saputra, "Analisis Perbandingan Hasil Pengolahan Citra Asli Dan Cropping Untuk Mengidentifikasi Karakteristik Tanaman Selada Menggunakan Metode Morfologi Dan Ekstrasi Ciri," *Jurnal Ilmiah SINUS*, vol. 21, no. 1, p. 73, Jan. 2023, doi: 10.30646/sinus.v21i1.664.

[15] R. Archana and P. S. E. Jeevaraj, "Deep learning models for digital image processing: a review," *Artif. Intell. Rev.*, vol. 57, no. 1, Jan. 2024, doi: 10.1007/s10462-023-10631-z.

[16] F. Sultana, A. Sufian, and P. Dutta, "A Review of Object Detection Models based on Convolutional Neural Network," Oct. 2019, doi: 10.1007/978-981-15-4288-6_1.

[17] T. Bariyah and M. Arif Rasyidi, "Convolutional Neural Network Untuk Metode Klasifikasi Multi-Label Pada Motif Batik Convolutional Neural Network for Multi-Label Batik Pattern Classification Method." *Jurnal Teknologi Informasi*, vol. 20, no. 1, 2021, doi: 10.33633/tc.v20i1.4224

[18] C. Diao, D. Kleyko, J. M. Rabaey, and B. Olshausen, "Generalized Learning Vector Quantization for Classification in Randomized Neural Networks and Hyperdimensional Computing," Feb. 2021. doi: 10.1109/IJCNN52387.2021.9533316.

[19] I. G. M. W. K. Widiantara, K. Y. E. Aryanto, and I. M. G. Sunarya, "Application of the Learning Vector Quantization Algorithm for Classification of Students with the Potential to Drop Out," *Brilliance: Research of Artificial Intelligence*, vol. 3, no. 2, pp. 262–269, Nov. 2023, doi: 10.47709/brilliance.v3i2.3155.

[20] K. M. Hosny, W. M. El-Hady, F. M. Samy, E. Vrochidou, and G. A. Papakostas, "Multi-Class Classification of Plant Leaf Diseases Using Feature Fusion of Deep Convolutional Neural Network and Local Binary Pattern," *IEEE Access*, vol. 11, pp. 62307–62317, 2023, doi: 10.1109/ACCESS.2023.3286730.

[21] . S. and M. Kaur, "A Comparative Analysis of Local Binary Pattern(LBP) Variants for Image Tamper Detection," Nov. 17, 2023. doi: 10.21203/rs.3.rs-3608580/v1.

[22] D. Valero-Carreras, J. Alcaraz, and M. Landete, "Comparing two SVM models through different metrics based on the confusion matrix," *Comput. Oper. Res.*, vol. 152, Apr. 2023, doi: 10.1016/j.cor.2022.106131.